MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2017/2018

TDS2111 – DATA STRUCTURES AND ALGORITHMS

(All Sections/Groups)

14 OCTOBER 2017
9:00am – 11:00am
(2 Hours)

INSTRUCTIONS TO STUDENTS

1. This question paper consists of 7 pages (including cover page) with 4 structured questions.
2. Answer ALL questions. The distribution of the marks for each question is given.
3. Please write all your answers in the answer booklet provided.

Question 1

(a) Write a recursive function `findMax()` to find the highest value in an array `y`. Note that the array `y` consists of six integers between indices `low` and `high`. The recursive function `findMax()` should consist of a template function `max()`.

(2 marks)

Answer:

```
#include <iostream>
#include <algorithm>

using namespace std;

int findMax(int y[], int low, int high)
{
// TODO: Fill in your code(s) here. Write the code(s) in
   answer booklet.




}

int main(void)
{
  int low = 0;
  int high = 5;
  int y[6] = { 2, 11, 10, 29, 4, 23 };
  cout << "The biggest number is "<<findMax(y,low,high)<<endl;
  return 0;
}
```

Sample Output:

```
The biggest number is 29.
```

Continued.....

(b) A function `EvaluatePostfix()` is quoted from a program and is partially printed out on this examination paper. The program is used to evaluate a postfix expression and return an output. Write the codes to complete `EvaluatePostfix()`.

(3 marks)

```
int EvaluatePostfix(string math_exp)
{
      // Assume <Stack> has been included in the program.
      // Declare a Stack from Standard template library in C++.
      // TODO: Fill in your code(s) here. Write the code(s) in
         answer booklet.


      for(int i = 0; i < math_exp();i++) {
      // Scan each character from left.
      // If character is a delimiter, move on.
      If(math_exp[i] == ' ') continue;

      // If character is an operator, pop two elements from stack,
      // perform operation and push the result back.
      Else if(IsOperator(math_exp[i])) {
            // Retrieve two operands from stack.
            // Assume they are operand1 and operand2 of datatype int
            // TODO: Fill in your code(s) here. Write the code(s)
               in answer booklet.


            // Perform operation
            int result = Operation(math_exp[i], operand1, operand2);

            // Assign the result of operation on stack.
            // TODO: Fill in your code(s) here. Write the code(s)
               in answer booklet.

      }

      // the rest codes of this function are not shown.
}
```

Continued.....

(c) The following program shows a portion of a pointer-based implementation of Queue. Write the code to complete the `enqueue()`. In this `enqueue()`, if the queue is empty, add the first item in the queue. However, if the queue is not empty, add a new node in the queue.

(5 marks)

```
typedef char QueueItemType;

Class Queue
{
 public:
 //constructors and destructor
  Queue();
  Queue(const Queue& Q);
  ~Queue() ;
 //Queue operations
  bool isEmpty() const ;

  void enqueue(QueueItemType newItem);
  void dequeue();
  void dequeue(QueueItemType& queueFront);
  void getFront(QueueItemType& queueFront) const;
  void display() ;

 private :
  struct QueueNode
    {QueueItemType item ;
     QueueNode *next ;} ;
  QueueNode *frontPtr ;
  QueueNode *backPtr ;
  QueueNode *curPtr ;
} ;

void Queue::enqueue(QueueItemType newItem)
{
  // TODO: Fill in your code(s) here. Write the code in answer
     booklet.









}
```

Continued.....

Question 2

(a) Use selection sort to sort the following list in ascending order. Show the initial list and the subsequent lists after each swap. Indicate the selected element and the elements in order after each swap.

31, 9, 16, 42, 12

(5 marks)

(b) Given the following list

2, 23, 37

and the primary and secondary hash functions:

h1(key) = key mod 7

h2(key) = 5 – (key mod 5)

Show the steps and store the items on the list into an array of size equal to 7 using double hashing.

(3 marks)

(c) Given an integer x > 0, write a recursive function *countUp* that writes the integers 1,2,..., x. (2 marks)

```
#include <iostream>
#include <cstdlib>

using namespace std;

void countUp(int);

int main()
{
   int x;
   cout << "Enter an integer: ";
   cin >> x ;

   countUp(x);
   system("pause");
   return 0;
}

void countUp(int x)
{
     // TODO: Write your codes in the answer booklet.
}
```

Continued.....

Question 3

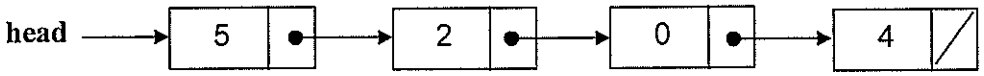
(a) Refer to the given structure of linked list node for C++ codes and the respective visualization of the linked list implementation.

(3 marks)

```
struct Node
{
    int data;
    Node *next;
};
```

```
void operation(Node* head)
{
  if (head == NULL)
    return;

  operation (head->next);
  cout << head->data;
}
```



If `void operation(Node* head)` is being called in the main function, explain the output of the linked list *before* and *after* the execution.

(b) Beginning with an empty binary search tree,

(i) Create a binary search tree when you insert the following values in an increasing order. (5 marks)

`8, 10, 14, 3, 6, 7, 1, 13, 4`

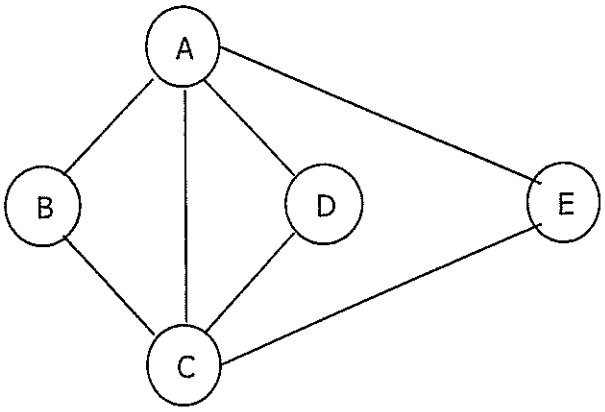
(ii) What is the height of the binary search tree? (0.5 mark)

(iii)Explain whether the tree is full or complete or balanced? (1.5 marks)

Continued.....

Question 4

The following undirected graph is provided:



(a) Write down its adjacency matrix representation and adjacency list representation. (2 marks)

(b) Performing Depth-First-Search (DFS) on a connected graph produces a spanning tree. There are 6 possible spanning trees when we start DFS from B. Draw all the 6 spanning trees. (6 marks)

(c) Performing Breadth-First-Search (BFS) on a connected graph produces a spanning tree. There are 2 possible spanning trees when we start BFS from B. Draw both spanning trees. (2 marks)

End of Page